# Freeform Search

**Database:**
US Pre-Grant Publication Full-Text Database
US Patents Full-Text Database
US OCR Full-Text Database
EPO Abstracts Database
JPO Abstracts Database
Derwent World Patents Index
IBM Technical Disclosure Bulletins

**Term:** L6 and failure

**Display:** 50 **Documents in Display Format:** FRO **Starting with Number** 1

**Generate:** ○ Hit List ● Hit Count ○ Side by Side ○ Image

Search    Clear    Interrupt

---

## Search History

**DATE:** Wednesday, November 24, 2004    Printable Copy    Create Case

| Set Name | Query | Hit Count | Set Name |
|----------|-------|-----------|----------|
| side by side | | | result set |
| | DB=USPT; PLUR=YES; OP=OR | | |
| L9 | 5734859.pn. | 1 | L9 |
| L8 | L7 and staging | 1 | L8 |
| L7 | L6 and failure | 4 | L7 |
| L6 | L5 and (restor$) | 4 | L6 |
| L5 | L4 and (parity near group) | 5 | L5 |
| L4 | L3 and (manag$ near redundant) | 24 | L4 |
| L3 | redundant near data | 3721 | L3 |
| L2 | L1 and redundant | 0 | L2 |
| L1 | 6212524.pn. | 1 | L1 |

END OF SEARCH HISTORY

L7: Entry 2 of 4                          File: USPT                    Sep 28, 1999


DOCUMENT-IDENTIFIER: US 5959860 A
TITLE: Method and apparatus for operating an array of storage devices

Abstract Text (1):
A storage controller operates an array of parity protected data storage units as a RAID level
5. One of the storage units is a dedicated write assist unit. The assist unit is a temporary
storage area for data to be written to the other units. When the array controller receives data
from a host, it first writes the data to the assist unit. Because the assist unit is not parity
protected and is only temporary storage, it is possible to write data to the assist unit
sequentially, without first reading the data, greatly reducing response time. The array
controller signals the CPU that the data has been written to storage as soon as it has been
written to the assist unit. Parity in the array is updated asynchronously. In the event of
system or storage unit failure, data can be recovered using the remaining storage units and/or
the assist unit. The write assist unit also doubles as a spare unit. Data recovered from a
failed unit can be stored on the write assist, which then ceases to function as a write assist
unit and assumes the function of the failed storage unit.

Brief Summary Text (4):
The extensive data storage needs of modern computer systems require large capacity mass data
storage devices. A common storage device is the magnetic disk drive, a complex piece of
machinery containing many parts which are susceptible to failure. A typical computer system
will contain several such units. The failure of a single storage unit can be a very disruptive
event for the system. Many systems are unable to operate until the defective unit is repaired
or replaced, and the lost data restored.

Brief Summary Text (11):
A single parity block of a RAID-3, RAID-4 or RAID-5 provides only one level of data redundancy.
This ensures that data can be recovered in the event of failure of a single storage unit.
However, the system must be designed to either discontinue operations in the event of a single
storage unit failure, or continue operations without data redundancy. If the system is designed
to continue operations, and a second unit fails before the first unit is repaired or replaced
and its data reconstructed, catastrophic data loss may occur. In order to support a system that
remains operational at all times, and reduces the possibility of such catastrophic data loss,
it is possible to provide additional standby storage units, known as "hot spares". Such units
are physically connected to the system, but do not operate until a unit fails. In that event,
the data on the failing unit is reconstructed and placed on the hot spare, and the hot spare
assumes the role of the failing unit. Although the hot spares technique enables a system to
remain operational and maintain data redundancy in the event of a device failure, it requires
additional storage units (and attendant cost) which otherwise serve no useful function.

Brief Summary Text (14):
Another object of this invention is to provide an enhanced method and apparatus for managing a
redundant array of storage devices in a computer system.

Brief Summary Text (21):
The storage management mechanism maintains status information in the array controller's memory
concerning the current status of data being updated. The amount of memory required for such
status information is relatively small, much smaller than the data itself. This status
information, together with the write assist unit, provide data redundancy at all times. In the
event of a failure of the assist unit, the array controller continues to update data from the

contents of its RAM as if nothing had happened. In the event of a _failure_ of a storage unit in the array other than the assist unit, the data on that unit can be reconstructed using the remaining units in the array (including the assist unit) and the status information. Finally, in the event of _failure_ of the controller itself, the storage units (including the assist unit) contain information needed for complete recovery.

Brief Summary Text (22):
The write assist unit also doubles as a spare unit in the event of _failure_ of another unit in the array. After any incomplete write operations are completed and parity updated, the data in the failed storage unit is reconstructed by Exclusive-ORing all the other units, and this data is stored on the assist unit. The assist unit then ceases to function as an assist unit, and functions as the failed storage unit that it replaced. The system then continues to operate normally, but without a write assist unit. The only effect is that data updates cause a greater impact to system performance, but data is otherwise fully protected.

Drawing Description Text (9):
FIG. 8 is a high-level flow diagram showing the steps taken by the array controller in the event of _failure_ of one of the service disk units, according to the preferred embodiment;

Drawing Description Text (10):
FIG. 9 shows the steps required to complete any incomplete write operations in the event of _failure_ of one of the service disk units, according to the preferred embodiment;

Detailed Description Text (9):
Memory 202 contains several records which support operation of the write assist unit in accordance with the preferred embodiment. Uncommitted list 212 in dynamic RAM 203 is a list representing those WRITE operations which may be incomplete. In particular, after array controller 103 receives a WRITE command from host 101, writes the data to write assist unit 104, and signals the host that the operation is complete, there will typically be some time delay before the data is actually written to the service units 105-108 and parity updated. Uncommitted list 212 records those operations which may be in such a pending status. If a device_failure should occur before the data can be written to the service units and parity updated, uncommitted list 212 will be used for recovery, as described more fully below. In the preferred embodiment, uncommitted list 212 is a variable length list of addresses on assist unit 104 at which the respective incomplete WRITE operations have been stored.

Detailed Description Text (40):
The storage subsystem of the present invention is designed to preserve data in the event of _failure_ of any single disk unit or loss of contents of the array controller dynamic memory 204. In the former event, the subsystem can dynamically recover and continue operation. The latter event is generally indicative of a loss of system power or such other catastrophic event in which the system as a whole is affected. In this case, the integrity of data on the storage units is preserved, although the controller will not necessarily be able to continue operation until the condition causing the _failure_ is corrected.

Detailed Description Text (41):
From the perspective of array controller 103, each storage unit 104-108 is a self-contained unit which is either functioning properly or is not. The storage unit itself may contain internal diagnostic and error recovery mechanisms which enable it to overcome certain types of internal defects. Such mechanisms are beyond the scope of the present invention. As used herein, the _failure_ of a storage unit means _failure_ to function, i.e., to access data. Such a _failure_ may be, but is not necessarily, caused by a breakdown of the unit itself. For example, the unit could be powered off, or a data cable may be disconnected. From the perspective of the controller, any such _failure,_ whatever the cause, is a _failure_ of the storage unit. Detection mechanisms which detect such _failures_ are known in the art.

Detailed Description Text (42):
In the event of _failure_ of write assist unit 104, array controller 103 updates its status information in non-volatile RAM to reflect that the assist unit is no longer in service, and thereafter continues operation of the service units as before, without using the write assist

unit.

Detailed Description Text (43):
FIGS. 8 and 9 represent the steps taken by array controller 103 in the event a failure of one
of the service units 105-108 is detected. FIG. 8 is a high-level flow diagram of the overall
recovery process. The controller first deactivates the write assist function so that no more
WRITE commands are written to the write assist unit at step 801. The controller then completes
the writing of any incomplete WRITE operations in its uncommitted list 212 to the service
units, including the updating of parity, at step 802. The controller then dynamically reassigns
storage space previously allocated to the failed service unit to the write assist unit at step
803. Data on the failed service unit is then reconstructed by Exclusive-ORing the data at the
same location on the remaining service units, and saved on the unit formerly allocated as the
write assist unit, at step 804. There may be some overlap of steps 802-804. The subsystem then
continues normal function without write assist, with the write assist unit 104 performing the
function of the failed service unit, at step 805.

Detailed Description Text (44):
FIG. 9 illustrates the steps required to complete any incomplete WRITE operations, which are
represented in FIG. 8 by the single block 802. There are several possible cases, each of which
requires individual consideration. If the incomplete write operation does not require any
further access to the failed service unit (step 901), then the write operation proceeds
normally at step 904. This would be the case either where the write operation never required
access to the failed unit, or where the failed unit had already been accessed prior to its
failure. If access is required, but no read access is required (i.e., only write access is
required, step 902), then the controller simply omits the write to the failed disk unit, and
otherwise continues the write operation normally as if the failed unit had been written to at
step 905. This would be the case, for example, where steps 402,403 of FIG. 4 had been completed
prior to the disk unit failure, but where step 405 had not. It could also occur, for example,
where a write operation involves all or nearly all of the blocks on a single stripe, and
instead of reading each block before writing to produce a change mask as shown in FIG. 4, the
blocks are either read only or written to only, and a change mask accumulated with each read or
write, as described above.

Detailed Description Text (49):
In the event of loss of the contents of controller memory, the data to be written, as well as
the list of incomplete write operations, will be contained in the write assist unit 104. After
controller operation is restored, the controller locates the most recent uncommitted list on
the write assist unit, loads this list into its dynamic memory, and performs each write
operation on the list to make the storage subsystem current. Because the most recent
uncommitted list on the write assist unit is not necessarily updated each time a write
operation completes, it is possible that some write operations on the uncommitted list will
have already completed. However, rewriting this data will not affect data integrity.

Detailed Description Text (53):
In the preferred embodiment, a single array controller services a plurality of disk drives in a
storage subsystem. The disk drives themselves are redundant, enabling the subsystem to continue
operation in the event of failure of a single drive, but the controller is not. Alternatively,
it would be possible to operate the storage subsystem with multiple redundant controllers,
enabling the system to remain operational in the event of failure of any single controller.
Because the write assist unit maintains data redundancy, it would not be necessary for the
multiple controllers to contain redundant uncommitted lists, command queues, and other data.
For example, assuming proper physical connections exits, it would be possible to operate a
subsystem having controllers A and B, in which controller A services disk drives 1 to N, and B
services disk drives (N+1) to 2N. In the event of failure of any one controller, the other
would service all disk drives 1 to 2N, using the information in the write assist unit to
recover incomplete write operations. In this case, the subsystem would continue to operate
despite the failure of a single controller, although its performance may be degraded.

Detailed Description Text (54):
In the preferred embodiment, a single write assist unit is associated with a single parity

group of service units (i.e., a group of service units which share parity). However, it would alternatively be possible to operate a storage subsystem according to the present invention with multiple write assist units. Additionally, it would be possible to operate a subsystem having multiple parity groups, in which one or more write assist units are shared among the various parity groups of service units.

CLAIMS:

1. A storage subsystem for a computer system, comprising:

a storage subsystem controller, said controller having a processor and a memory;

at least four data storage units coupled to said controller, wherein at least one of said data storage units is a write assist data storage unit, and at least three of said data storage units are service data storage units;

at least one set of storage blocks, each set comprising a plurality of data storage blocks for containing data and at least one data redundancy storage block for containing data redundant of the data stored in said data storage blocks, each of said storage blocks of a set being contained on a respective service data storage unit;

means in said controller for maintaining said data redundancy storage block on said set of storage blocks;

means in said controller for receiving write data, said write data being data to be·written to said data storage units, said write data being contained in a plurality of write commands;

selection means, responsive to said means in said controller for receiving write data and operable when sufficient available storage space exists on said write assist unit to store data contained in a write command, for selectively determining with respect to individual ones of said write commands whether said received write data should be written to said write assist unit;

means for writing said write data to said write assist unit, wherein said means for writing said write data to said write assist unit selectively writes said write data to said write assist unit in response to said determination made by said selection means;

means in said controller for signalling operation complete after writing said data to said write assist unit and before writing said data to any of said service data storage units;

means for reconstructing said data in the event any one of said data storage units fails after signalling operation complete; and

means for reconstructing said data in the event the contents of said memory are lost. after signalling operation complete.

8. A storage apparatus for a computer system, comprising:

a write assist data storage unit;

a plurality of service data storage units;

means for maintaining data redundancy among said plurality of service data storage units;

means for receiving write data, said write data being data to be written to said plurality of service data storage units, said write data being contained in a plurality of write commands;

selection means, operable when sufficient available storage space exists on said write assist unit to store data contained in a write command, for selectively determining, with respect to individual ones of said write commands, whether said write data should be temporarily stored in

said write assist unit;

means for temporarily storing said write data in said write assist unit, wherein said means for temporarily storing said write data in said write assist unit selectively writes data to said write assist unit in response to said determination made by said selection means;

means for reconstructing data stored on a service data storage units in the event of <u>failure</u> of said unit; and

means for storing said reconstructed data on said write assist unit.

10. The storage apparatus of claim 8, further comprising:

means for disabling the write assist function of said write assist unit in the event of <u>failure</u> of a service data storage unit; and

means for operating said write assist unit as said service unit which failed.

14. A method for storing data in a computer system, comprising the steps of:

storing data redundantly on a plurality of service data storage units;

selectively determining whether updated data to be written to said plurality of service units should be written to a write assist data storage unit, said updated data being contained in a plurality of write commands, said selectively determining step being performed with respect to individual ones of said plurality of write commands and when sufficient available storage space exists on said write assist unit to store the updated data contained in the respective individual write command;

writing said updated data to said write assist data storage unit, said writing step being performed in response to said selectively determining step determining that said updated data should be written to said write assist unit;

signalling that said updated data has been written to said plurality of service data storage units;

writing said updated data redundantly to said plurality of service data storage units, wherein said step of writing said updated data to said plurality of service data storage units is completed after said signalling step;

reconstructing data stored in a service data storage unit in the event of <u>failure</u> of said service data storage unit; and

storing said reconstructed data on said write assist unit, and thereafter operating said write assist unit as said service unit which failed, in the event of said <u>failure</u> of said service data storage unit.

18. A storage subsystem controller for a computer system, comprising:

a processor;

a memory;

a host interface for communicating with a host computer system;

a storage unit interface for communicating with at least four data storage units coupled to said controller, wherein at least one of said data storage units is a write assist data storage unit, and at least three of said data storage units are service data storage units,

wherein said service data storage units comprise at least one set of storage blocks, each set

comprising a plurality of data storage blocks for containing data and at least one data redundancy storage block for containing data redundant of the data stored in said data storage blocks, each of said storage blocks of a set being contained on a respective service data storage unit;

means for maintaining said data redundancy storage block on said set of storage blocks;

means for receiving write data from said host computer system, said write data being data to be written to said data storage units, said write data being contained in a plurality of write commands;

selection means, responsive to said means for receiving write data from said host computer system and operable when sufficient available storage space exists on said write assist unit to store data contained in a write command, for selectively determining with respect to individual ones of said plurality of write commands whether said received write data should be written to said write assist unit;

means for writing said write data to said write assist unit, wherein said means for writing said write data to said write assist unit selectively writes said write data to said write assist unit in response to said determination made by said selection means;

means for signalling operation complete to said host computer system after writing said data to said write assist unit and before writing said data to any of said service data storage units;

means for reconstructing said data in the event any one of said data storage units fails after signalling operation complete; and

means for reconstructing said data in the event the contents of said memory are lost after signalling operation complete.